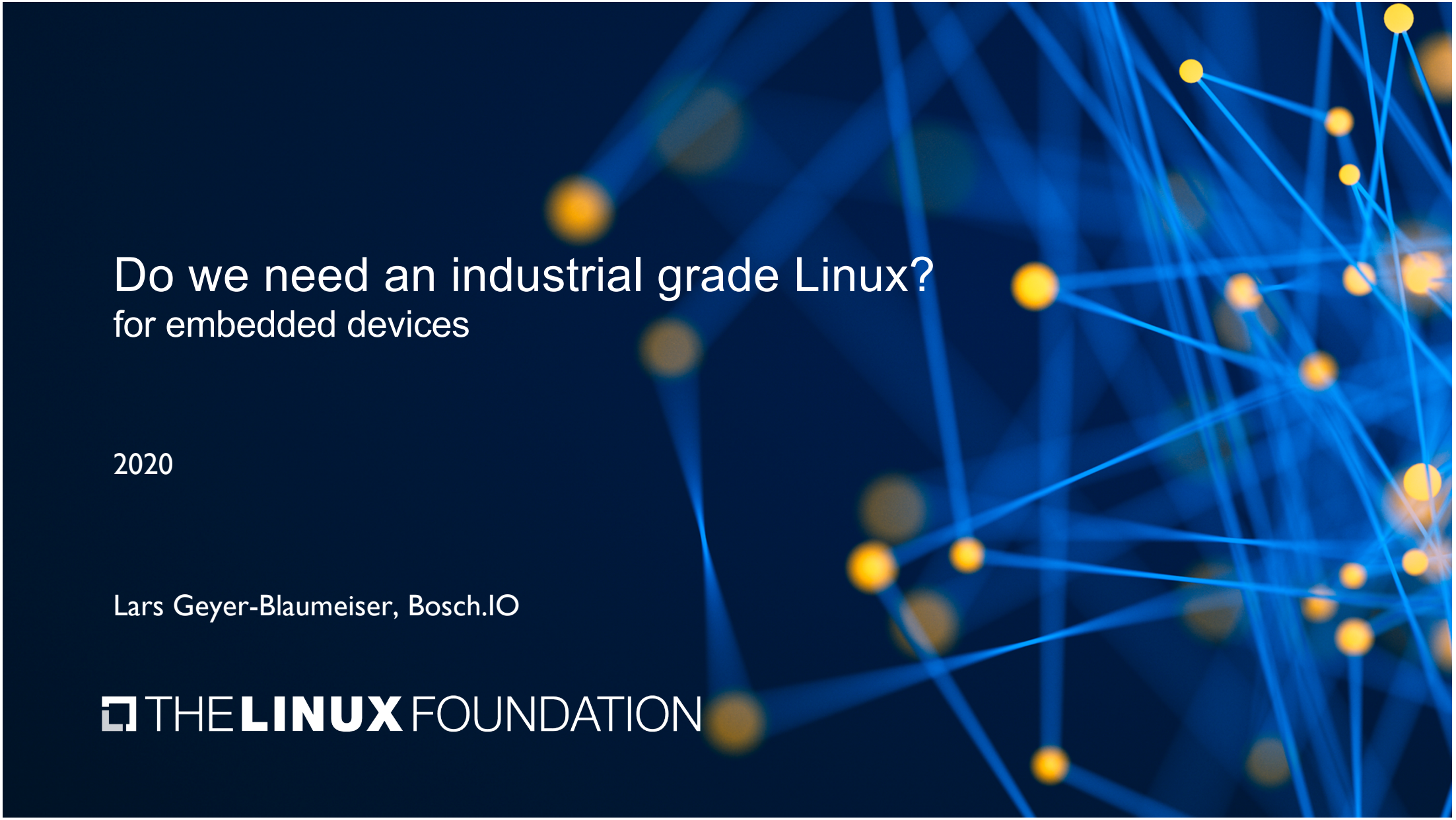


Do we need an industrial grade Linux?

for embedded devices

2020

Lars Geyer-Blaumeiser, Bosch.IO

THE LINUX FOUNDATION

Holistic approach for product diversity based on Linux

Project Mission Statement

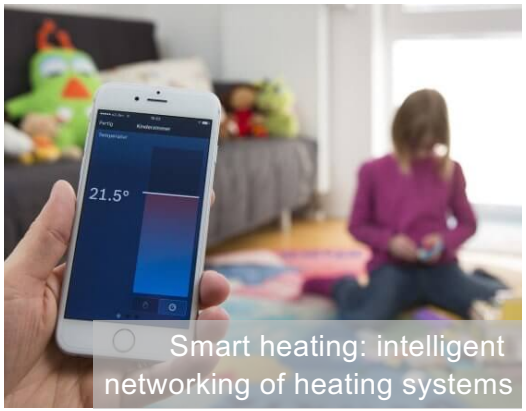
A versatile open source infrastructure originally tailored to the automotive connectivity needs and fit for a wide variety of electronic devices. Security and modularity are two of its primary strengths.

Apertis provides a feature-rich framework supporting add-on software and resilient upgrade capabilities. Beyond an operating system, it offers new APIs, tools and cloud services.

We connect every thing

Bosch IoT Suite

Device management for connected filters



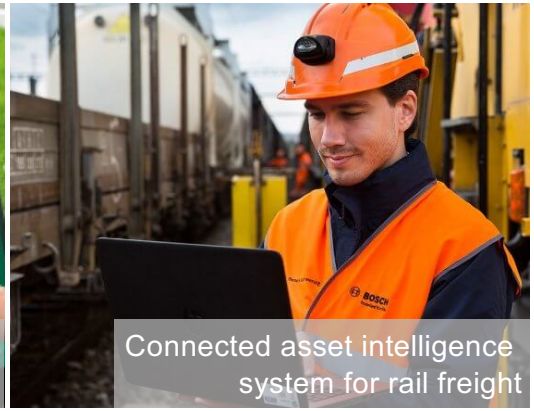
Smart heating: intelligent networking of heating systems


IoT Gateway: easy connection to Industry 4.0 environments


'Internet of Oysters': monitor the health of the water system

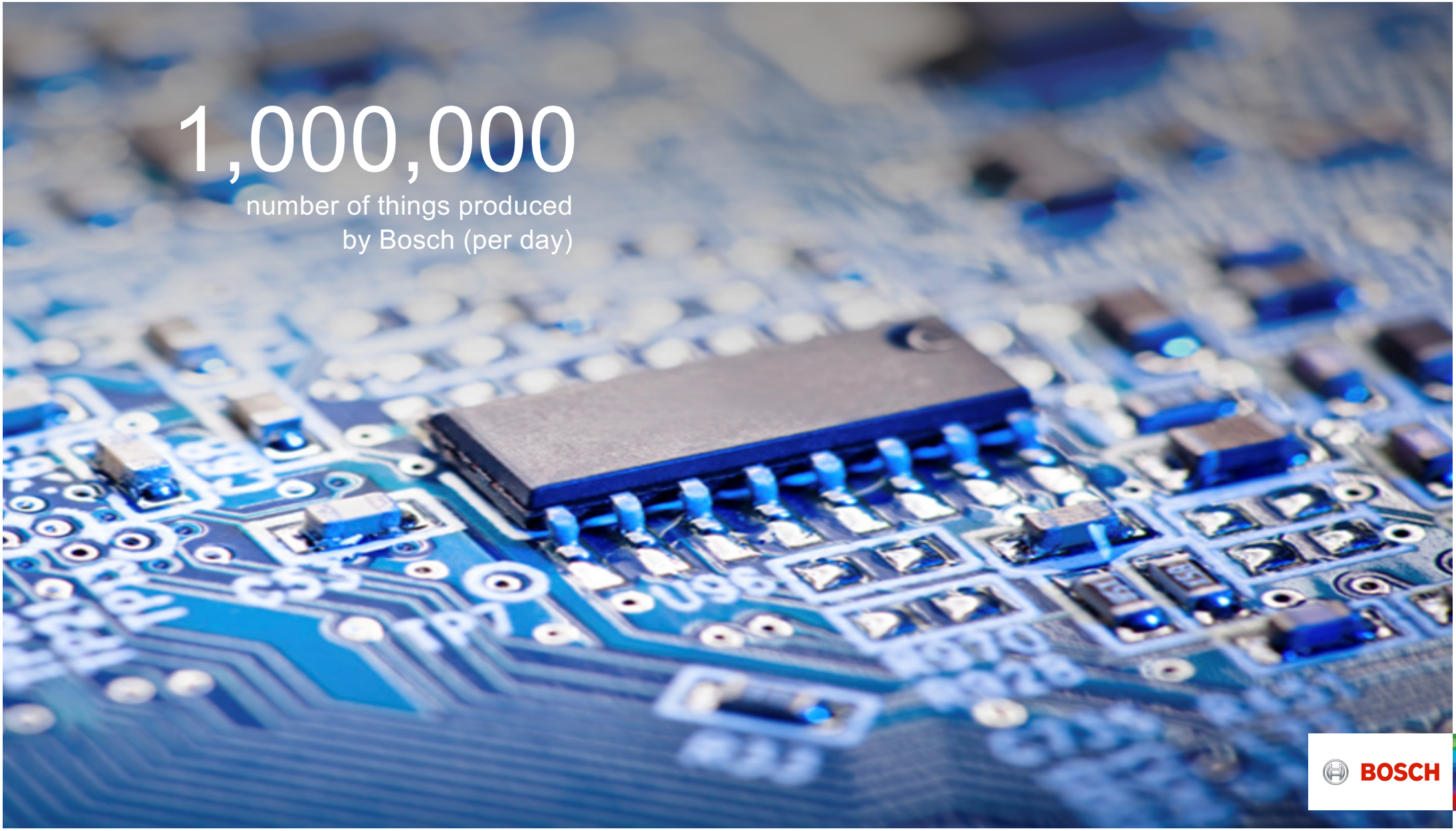

TrackMyTools: a smart inventory management system


Indego Connect: intelligent device management


Connected asset intelligence system for rail freight


Firmware and software updates over the air

BOSCH


1,000,000
number of things produced
by Bosch (per day)
BOSCH

"In a few years, every electronic product will be internet-capable. The question is no longer if, but when."

Dr. Volkmar Denner

Chairman, Board of Management
Robert Bosch GmbH

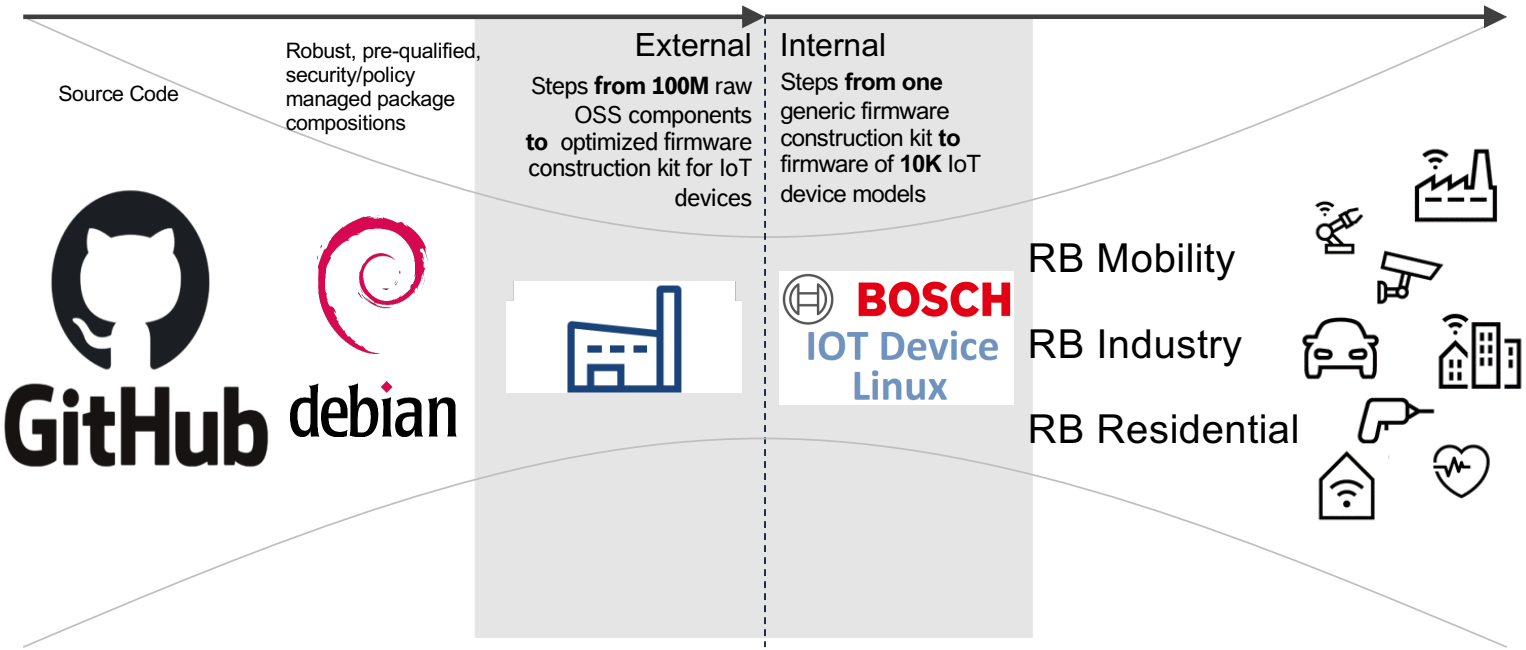
Typical Demand of Industrial Devices

- Today systems evolve: Development lasts until end of product usage
- Products are integrated into systems, e.g., an IoT service
- Time-to-market is rather short these days
  - Efficiency of development
  - Collaboration in ecosystems where new trends are magically prepared
- Blurring boundaries for products
  - Specialized products for specific use cases
  - New services on top of existing products

Typical Demand of Industrial Device Software

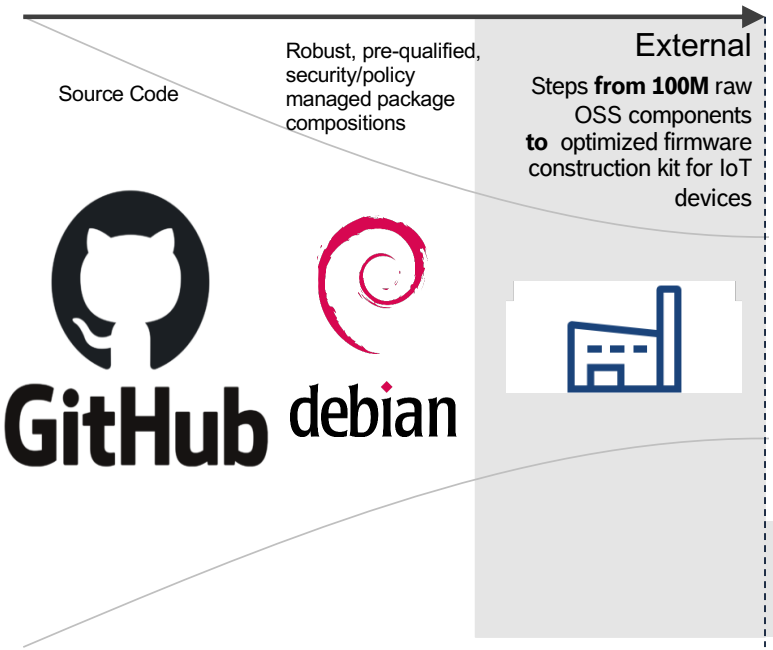
- Optimized for specific hardware
- Optimized for specific functionality
- Secure, reliable, simplified user interface, perfect integration into relevant environment, easy to use plug & play connectivity
- Hardening against any kind of attack
- Integration of commercial and proprietary components
- Life-cycle management of used components
- Secure & reliable update system
- Reuse of infrastructure and general components
- App concept with two level of trust in the software stack (base system & applications)

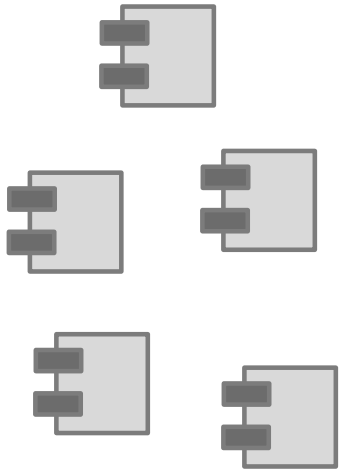
Harvesting best OSS for IoT device Linux



Approach: from variety of OSS artifacts – to a tailored distribution – for many IoT devices

Harvesting best OSS for IoT device Linux



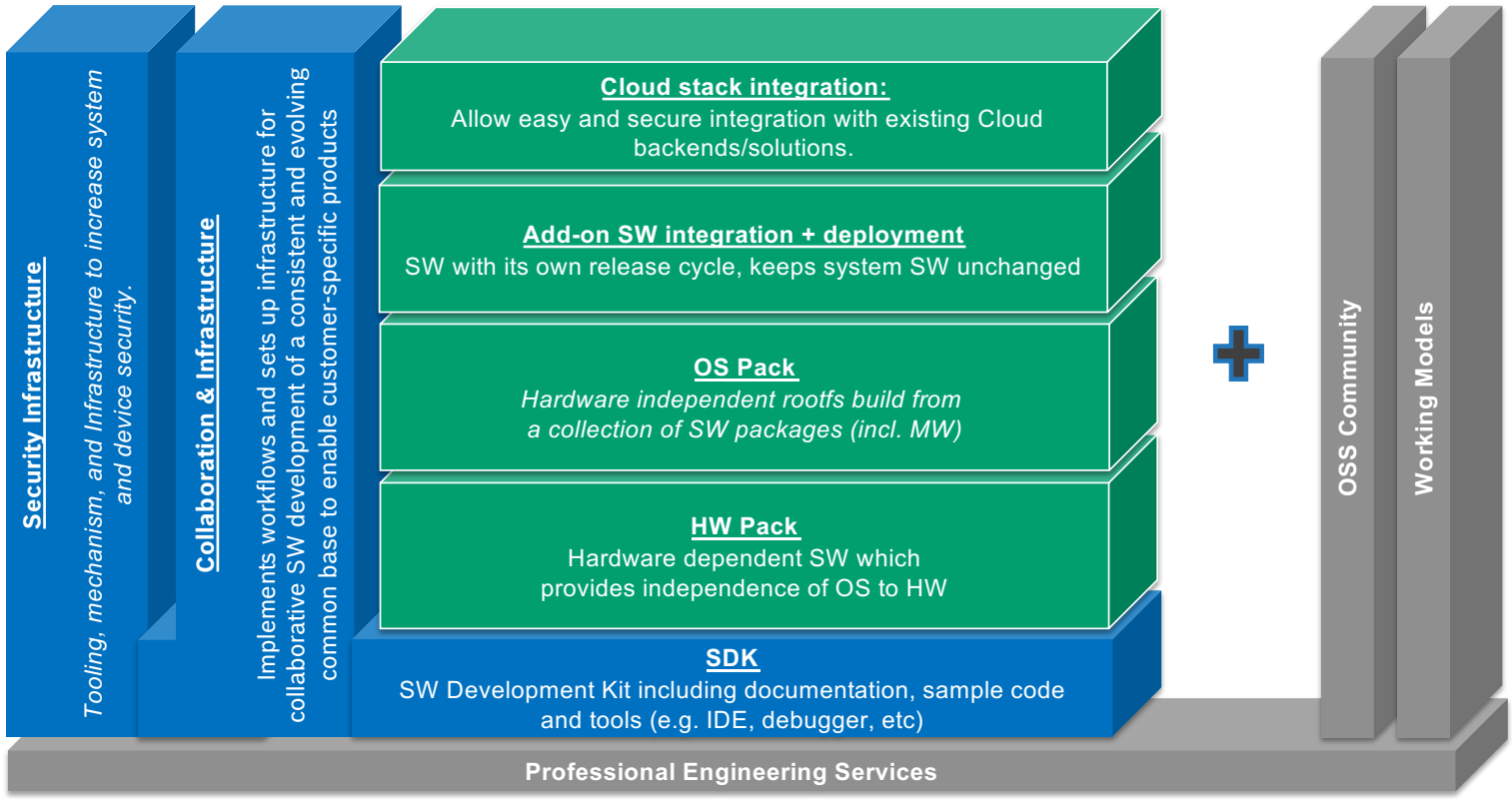


Industrial Strength Components

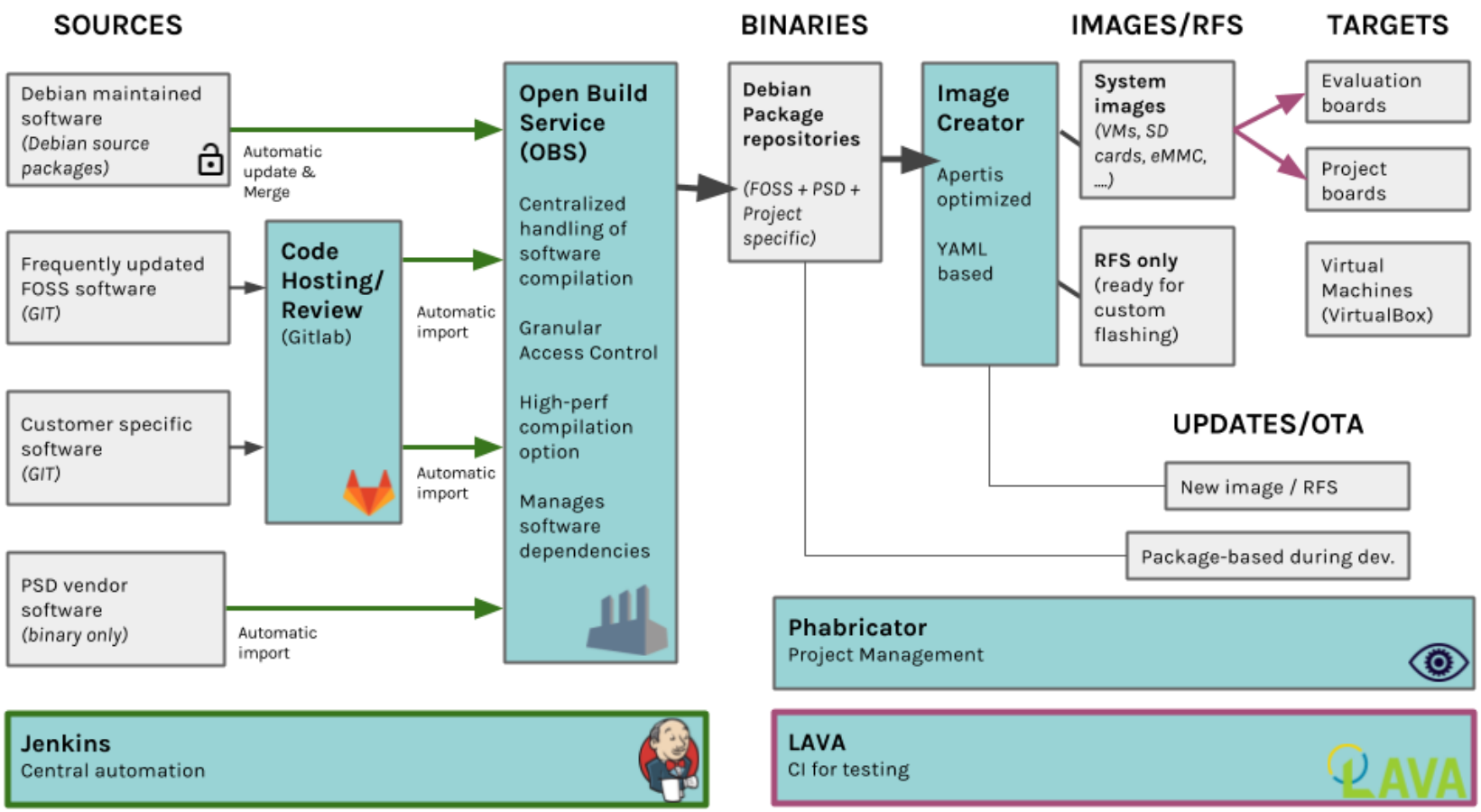
- Source or binary
- Tested in reference systems
- S-BOM enabled
- Open source metadata
- Vulnerability data and mgmt
- Ready to use
- Prepared for long term support

Seamless infrastructure extension

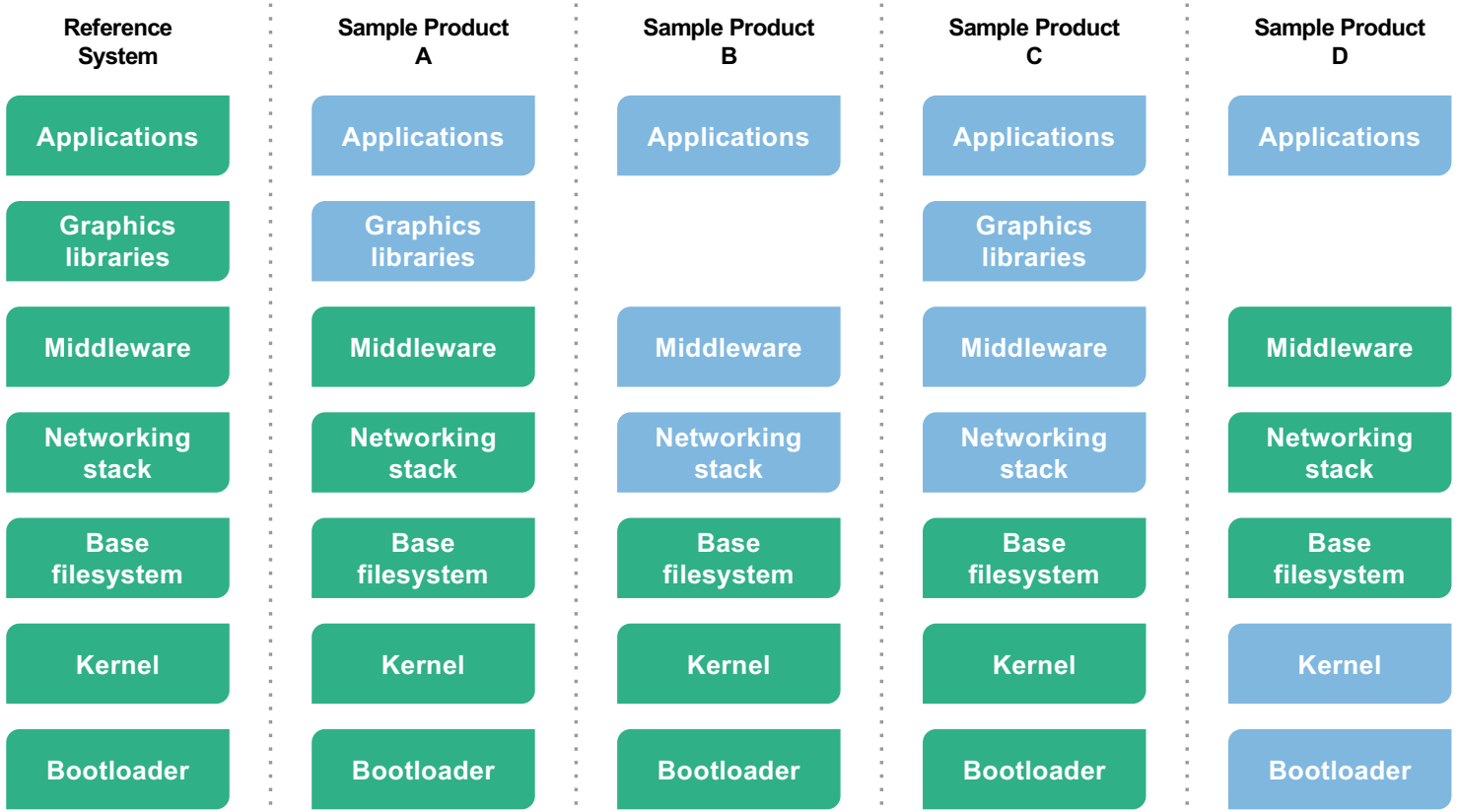
Aspects of the project



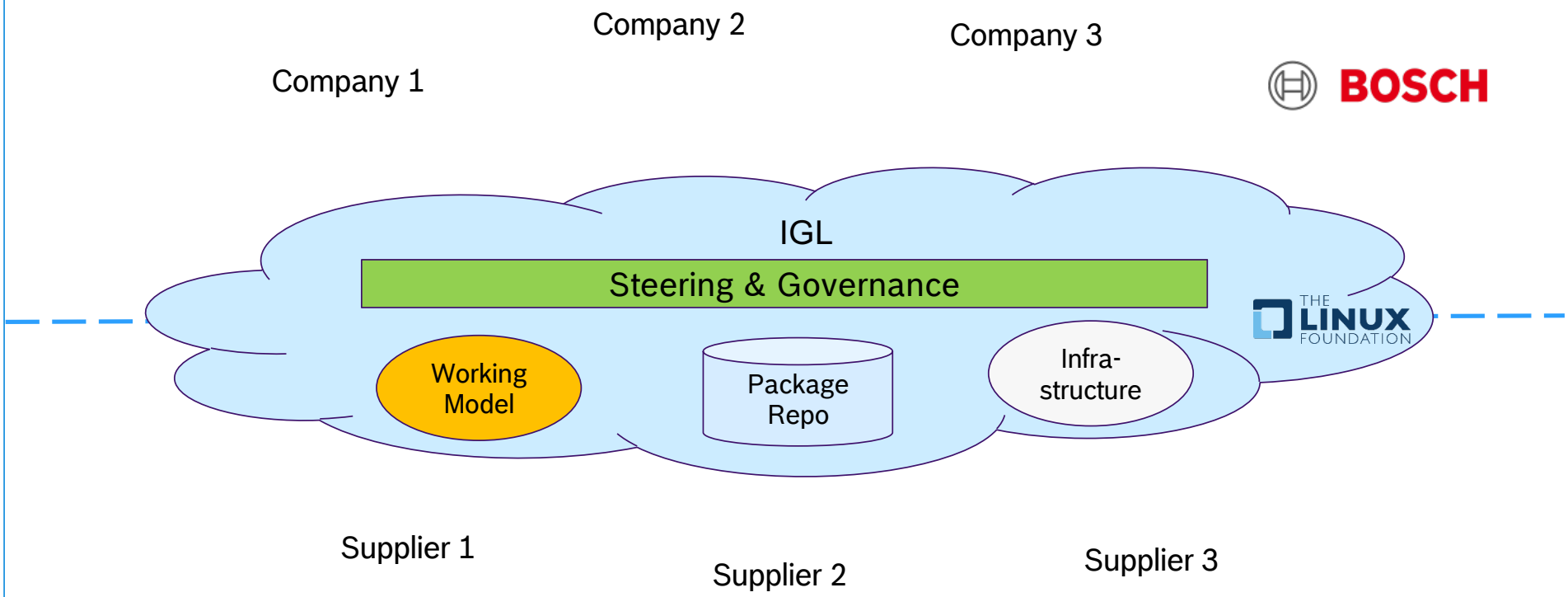
Provided Infrastructure



Modularity


Reference System
Applications
Graphics libraries
Middleware
Networking stack
Base filesystem
Kernel
Bootloader
Sample Product A
Applications
Graphics libraries
Middleware
Networking stack
Base filesystem
Kernel
Bootloader
Sample Product B
Applications
Middleware
Networking stack
Base filesystem
Kernel
Bootloader
Sample Product C
Applications
Graphics libraries
Middleware
Networking stack
Base filesystem
Kernel
Bootloader
Sample Product D
Applications
Middleware
Networking stack
Base filesystem
Kernel
Bootloader

The Community



Governance Model

Control complexity

- Idea:
  - Get the non-competitive parts done together
  - Drive innovation in collaboration
- Provide a collaboration infrastructure to get industry strength in the provided packages
  - Enable maximal reuse by keeping absolute freedom for project specific solutions
  - Component validation on reference hardware
  - Life-cycle information like oss license information or vulnerabilities provided
  - Package based images to keep traceability throughout product lifetime and provide long term support
- Prevent big bang switch by providing modularity

Project Principles

- Re-Use – be integrative and choose available projects whereever possible as starting point
- Upstream first – contribute as much as possible to upstream
- Modular – shift focus to distribution packages as building blocks to compose SW-systems
- Diversity – foster specilization in compositons based on consistent building blocks in commonality
- Distributed Development model – utilize the features provided by Debian packaging system
- Efficient – minimize individual efforts by maximizing common scopes
- Device – optimized support for embedded systems

Thank you!

Legal Notices

The Linux Foundation, The Linux Foundation logos, and other marks that may be used herein are owned by The Linux Foundation or its affiliated entities, and are subject to The Linux Foundation's Trademark Usage Policy at https://www.linuxfoundation.org/trademark-usage, as may be modified from time to time.

Linux is a registered trademark of Linus Torvalds. Please see the Linux Mark Institute's trademark usage page at https://lmi.linuxfoundation.org for details regarding use of this trademark.

Some marks that may be used herein are owned by projects operating as separately incorporated entities managed by The Linux Foundation, and have their own trademarks, policies and usage guidelines.

All other trademarks are the property of their respective owners. Use of such marks herein does not represent affiliation with or authorization, sponsorship or approval by such owners unless otherwise expressly specified.

The Linux Foundation is subject to other policies, including without limitation its Privacy Policy at https://www.linuxfoundation.org/privacy and its Antitrust Policy at https://www.linuxfoundation.org/antitrust-policy, each as may be modified from time to time. More information about The Linux Foundation's policies is available at https://www.linuxfoundation.org.

Please email legal@linuxfoundation.org with any questions about The Linux Foundation's policies or the notices set forth on this slide.